NRL Report 8780

Accuracy of Software Development Activity Data: The Software Cost Reduction Project

L.J. CHMURA

Computer Science and Systems Branch
Information Technology Division

A.F. NORCIO

Computer Science Department
United States Naval Academy
Annapolis, MD

December 30, 1983

NAVAL RESEARCH LABORATORY
Washington, D.C.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 8780	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* ACCURACY OF SOFTWARE DEVELOPMENT ACTIVITY DATA: THE SOFTWARE COST REDUCTION PROJECT		5. TYPE OF REPORT & PERIOD COVERED Final report on an NRL problem June 1982 to June 1983 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) L. J. Chmura and A. F. Norcio		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 75-0199-0-3 61153N RR014-09-41
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research 800 N. Quincy Street Arlington, VA 22217		12. REPORT DATE December 30, 1983 13. NUMBER OF PAGES 29
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Software engineering
Self reporting
Software development
Data collection
Data accuracy

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report discusses the accuracy of self-reported programmer activity data and valid ways to analyze the data. We conducted a one-week experiment during which we sampled the activities of five software engineers working on the Software Cost Reduction (SCR) project at the Naval Research Laboratory. We compared the sampled data with activity data submitted "as usual" by the five engineers on weekly activity reports. The results indicate that the engineers

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. ABSTRACT (Continued)

report relatively accurate data with the activity report if they make notes of their activity, or if they promptly submit their reports. If engineers do not keep notes of their activities, prompt reporting is critical to ensure the accuracy of reported data. The results also indicate that ratios between activity categories are valid metrics of project activity.

CONTENTS

ACCURACY OF SOFTWARE DEVELOPMENT ACTIVITY DATA: THE SOFTWARE COST REDUCTION PROJECT

INTRODUCTION

Since 1978, Naval Research Laboratory (NRL) researchers have collected data on changes and personnel activity connected with the Software Cost Reduction (SCR) project at NRL. The purpose is to evaluate the effects of the software engineering methodologies being used. A long-standing concern has been the accuracy of the collected data. Validation of change report forms has shown that originally submitted data often are incorrect or incomplete (Basili and Weiss 1983). In this report, we present the results of work begun in the summer of 1982 to determine the accuracy of software development activity data.

The SCR project described in Appendix A is a redevelopment of the version 2 Operational Flight Program for the Navy's A-7E aircraft. The software methodology evaluation (SME) project is a separate project that was established to provide an objective evaluation of the methodologies used in the cost reduction project. Data on software changes and personnel activity are being collected. Since the start of the project in 1978, data on more than 30,000 hours of project activity have been collected on forms that SCR project personnel are supposed to complete weekly.

The activity data collection form used is the Weekly Activity Report (WAR), shown in Fig. 1. The boxes on the form represent project activities. The front page of the report form is primarily concerned with the recording of module development activity, where means as defined by Parnas (1972). Space is provided for project personnel to provide the names of modules below the first two levels that are listed. The back page of the WAR form is concerned with integration testing and miscellaneous activities. An instruction sheet explains the activity categorization scheme. A submitted report is rather sparse; typically, it has only a few boxes marked with the hours spent on particular activities during the week.

One problem with this data collection scheme is miscategorization of activity. No instruction sheet can unambiguously define how people should map their myriad activities into a relatively small number of activity boxes. SME researchers attempt to avoid this problem by frequently checking submitted activity reports and notifying engineers of subtleties of work categories.

A second problem is inaccuracy of the submitted data. For example, if a person reports 4 hours devoted to a specific activity, how confident can we be that it was 4 hours, not 2 or 8? Three things suggest that the Software Cost Reduction project activity data may be inaccurate in terms of the hours reported. First, the Software Cost Reduction and Software Methodology Evaluation projects are distinct. The weekly data that engineers are required to submit support the SME project, not the SCR project. Second, engineers are supposed to submit activity reports weekly, but this is not always done. Third, the report instruction sheet does not contain suggested ways for personnel to record their activities that would encourage accuracy in reporting. A survey of engineers shows that they complete the reports in many different ways and that most rely solely on their memory when reporting weekly activities. The results of this survey, made in the summer of 1982, are shown in Appendix B.

Resolving the problem of inaccuracy is critical to SME researchers because plans call for using the data to characterize software development efforts that use such software engineering principles as information hiding (Parnas 1972) and abstract interfaces (Parnas 1977). But, the resolution is also of more general importance; other software researchers are collecting software project activity data in the same

Manuscript approved September 7, 1983

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: ____________ Date: Friday, ____________

Activity Area	Activity Hours										
	Design				Pseudo Code			EC Code			Unit Testing
	Creating	Discussing	Reading	Formal Reviewing	Creating	Discussing	Reading	Creating	Discussing	Reading	
SOFTWARE STRUCTURES											
Module........											
Process........											
Uses Hierarchy........											
SOFTWARE MODULES											
Hardware-Hiding........											
Extended Computer....											
Device Interface.....											
Behavior-Hiding........											
Function Driver......											
Shared Services......											
Software Decision......											
Application Data Type											
Physical Model.......											
Data Banker..........											
System Generation....											
Software Utility.....											
Resource Monitor.....											

(See back)

(30 Nov 81, WD-7810a)

(front)

Integration/ Performance Test	Activity Hours			
	Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements..........................	
DIM Specifications....................	
Function Driver Function Specifications.	
Module Guide..........................	
Project Control.......................	
Travel................................	
Other:	
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

(back)

Fig. 1 — SCR Project Weekly Activity Report (WAR)

manner (Card et al. 1982). Also, there is an apparent lack of published results on the issues of software activity data accuracy and how to determine it. Our approach to measuring accuracy is described in the following section.

RESEARCH APPROACH

Goals And Hypotheses

There were four goals for the research.

- Determine how accurately project personnel record their SCR activities on WARs.
- Determine which analyses of reported activity data are valid and which are not, based on the determined accuracy of the data.
- Determine the pros and cons of collecting personnel activity data through the use of weekly activity reports.
- Identify a useful technique for validating the accuracy of personnel activity data.

Concerning the first goal, one of our conjectures was that project activity times, reported in hours, were grossly different from actual hours spent on activities. We believed this for several reasons. People are generally reluctant to track their activities in detail. The WAR existed to serve the purposes of the SME project, not the SCR project that actually employed the people. There was no obvious benefit to engineers for conscientiously completing activity reports, other than to avoid being hounded by data collectors for not doing so. Many project personnel relied on their memories to fill in their weekly reports. Although a good instruction sheet was available that described activity categorization, there were no suggestions for to achieve accurate reporting.

A second conjecture was that personnel who completed activity reports at the end of the week, (those who are *prompt*), would generally be more accurate than those who completed them a week or more late (those who are *tardy*). Exceptions would be those project personnel who recorded their activities hour-by-hour or activity-by-activity and who used these notes when they completed their reports.

Concerning the second goal, we believed that collected SCR activity data, even if inaccurate, could be usefully analyzed in terms of the ratios or percentages of effort for one activity with respect to others.

Experimental Approach

The research consisted of a one-week experiment in which personnel activity was recorded in an alternative, presumably more accurate, manner. Personnel continued to fill in WARs 'as usual,' thereby allowing the comparison of reported data with the alternatively collected activity data.

Activity Sampling

The use of some form of continual observation to gather accurate SCR activity data (e.g., silent observer, active observer, video monitoring) was rejected for three reasons. First, much software development activity is silent in nature; personnel often just think or write. In such situations, passive observation would not guarantee accurate activity data. Active observation, in which the observer is free to question a participant about what he or she is doing at any time, would interfere with the activity being observed. Second, such an approach would be at odds with our intention of not informing the participants that the true purpose of the experiment was to check the accuracy of activity reporting. Knowing the true purpose would color normal reporting practices. It would be better to tell them that we were experimenting with finding better or easier ways of recording activity than is possible with the currently used report form. It seemed clear that no one would ever believe that continual observation was better or easier. Third, insufficient resources were available for continually observing people who worked in different offices.

Other methods described by Mintzberg (1973, Appendix B), for example, Secondary Sources, Questionnaire and Interview, Critical Incident and Sequence of Episodes, and Diary were not carefully considered. But, in retrospect, the inherent weaknesses of these four methods and the reasons given above for rejecting all observation methods also would have caused us to reject these methods.

The only reasonable alternative seemed to be an activity sampling method, of which the WAR form itself is an example. To achieve greater accuracy than with the weekly report, we decided to shorten the sampling period from 1 week to 2 hours and to use active sampling. A sampler would visit each participant every two hours and ask for a summary of past activity; participants would not have to write down anything about their recent activity unless that was their custom.

The choice of a 2-hour sampling period was not arbitrary. We thought a shorter period might appear as an unrealistic alternative to the WAR, and therefore might reveal the true purpose of the experiment to the participants. We were also concerned that too frequent sampling of activity would allow the participants to be unusually accurate with the weekly reports that they would eventually have to submit. A longer period (e.g., 4 hours) seemed too long to ensure accurate recall by those participants who relied solely on their memories to record SCR activity.

The choice of active sampling was made to make the sampling approach for collecting activity data a realistic alternative to the WAR. Many engineers are reluctant to report their activity weekly; there would be even more reluctance to report every 2 hours.

Participants

As shown in Appendix B, engineers completed WARs in quite different ways in the summer of 1982. Some recorded their activity hour-by-hour or task-by-task and transferred this information to a report form at the end of each week. Others filled out a report at the end of the week from memory. Because one of our hypotheses was that accuracy would vary with differing approaches to completing activity reports, we wanted participants who completed reports in different ways. We also wanted participants who exhibited different work styles. For example, we wanted some who worked on only one or two activities in a week and others who worked on several activities.

We felt that the success of the experiment depended on participation by a representative group of engineers. Five of the 11 engineers identified in Appendix B made up our representative set: Engineers *A, B, E, H,* and *K*. We chose only these five for two reasons. First, we felt that having more than five participants might reveal the true purpose of our experiment. If we were only experimenting with a different way of collecting activity data, why would we need to use more than four or five persons? Second, these five all worked full time at NRL and thus were convenient to sample. Engineer *J* also worked full time at NRL, but he was the head of the SME project.

Preparation

The five participants attended a pre-experiment briefing. The head of the SME project reviewed the reasons for collecting SCR project activity data. He cited the problems that were being experienced with using the WAR form, and told the participants that SME personnel were studying better ways to collect such data. He asked for help in evaluating a periodic sampling method to collect activity data. The sampling method was described. The explanation given for continuing to fill in weekly reports as usual was that SME researchers were uncertain about the accuracy of the sample data. After the experiment was completed, many of the engineers voiced the suspicion that the real objective of the study was not to test an alternative method of collecting project activity data.

The participants were told that their project management approved of their participation. It was made clear that the experiment was not something that would reflect on their job performance evaluation. Indeed, they could end their participation in the experiment at any time they wished. At the end of the briefing, all five engineers gave verbal consent to participate in the experiment.

In preparation for the sampling, we chose two research assistants to be the samplers; one would be the primary sampler, the other a backup. Both were college students employed at NRL for the summer. They were chosen because they were familiar with the SCR project and WAR terminology. Also, because the samplers were research assistants, they were not threatening to the participants.

For 2 days prior to the start of the actual sampling, the samplers experimented with the sampling approach by sampling the SCR activities of the head of the SME project (who of course was not one of the participants). Sampling in person and by the telephone was tried. Procedures for dealing with situations such as a participant who could not be located for sampling at a required time were refined.

The samplers knew that the true purpose of the experiment was to measure the accuracy of WAR data, and they attended the pre-experiment briefing of the five participants.

Conduct

Sampling began Monday morning, 9 August 1982, and continued for 1 week. The last sample was made on Monday morning, 16 August 1982. The samplers attempted to poll each participant at 10 a.m., noon, 2 p.m., and 4 p.m. regarding their SCR activities since they were last polled. Twenty samples for each participant were thus possible.

The participants had the choice of being interviewed in person or by telephone. All preferred the personal interview. Interview time was kept short to avoid disturbing the participants; typical times ranged from 30 seconds to 3 minutes. Because few participants were available at the precise sampling times, the samplers used a window approach to adjust, for example, to the luncheon schedules of participants. If a participant could not be contacted within the 1-hour sampling window, the samplers collected the missed activity data during the next scheduled window.

Each participant's activities for each sample period were recorded on a separate WAR form. The number of samples collected for Engineer and were, respectively, 19, 18, 12, 16, and 13. At the end of the experiment, all samples for the individual participant were tallied on a single WAR, which then was compared to the submitted WAR (see Appendixes D and E).

Closeout

At the end of the experiment, participants submitted their weekly activity reports as usual. Engineers *A* and *B* submitted their forms on 17 August; Engineer *K* submitted his form on 31 August. After one or more customary requests went out to all SCR personnel to catch up on overdue WARs, Engineer *E* and Engineer *H* finally submitted their reports on 3 September and 10 September, respectively.

On 20 August, SME researchers interviewed Engineers *A, B* and *H* and asked them how they liked the alternative way of gathering activity data. Specifically, the engineers were asked to comment on the effects of the 2-hour sampling period. The true purpose of the experiment was not revealed during the interviews. Engineers *E* and *K* were interviewed individually on 14 September. The responses appear in Appendix C.

Three of the five participants felt that the sampling approach was irritating or disruptive. It was difficult to determine if this annoyance affected their usual submission of activity reports. Engineer *A* specifically said that as a result of the sampling he lost track of detail and thereby the activity report he submitted was less accurate than usual. This was a surprise, for we had believed that the sampling would tend to reinforce the memories of those people who filled out reports from memory, which Engineer *A* generally did.

Analyses of Data

Two kinds of analyses were performed. To determine the accuracy of WAR reporting, we compared the differences between sampled SCR project activity and reported activity. To determine whether it is valid to look at ratios of reported WAR activities (e.g., the ratio of design-creating activity to design-discussing activity), we compared the differences between ratios computed from sampled data and those computed from reported data. Two simple statistical techniques were used for both comparisons -- the paired t test and correlation coefficients (Dixon and Massey 1969).

In comparing sampled versus reported activity hours for each engineer, two conventions were followed. First, if activity was sampled for submodules of a module, but an engineer reported activity only for the module, we combined all activity to the module level and compared module-level activities. Second, if hours were *reported* in a particular category, but no hours were *sampled* for that category, we considered the sampled value to be zero hours. Likewise, if hours were sampled but not reported in a category, we considered the reported value to be zero hours.

Column 2 of Table 1 gives the mean differences between sampled activity hours and reported hours for each of the five participants and for all five combined. The statistical significance of these

Table 1 — Sampled vs Reported SCR Activities:
Mean Differences and Paired *t* Test Results

Engineer	Mean Difference *(hours)*	Computed *t*	Critical *t* *(0.05)*
A	−1.38	−1.78	2.35
B	0.18	0.37	1.78
E	−0.86	−0.29	1.94
H	−0.67	−1.51	2.02
K	−13.00	...	...
Combined	−1.14	−1.16	1.72

differences was checked via a paired *t* test. The results, which appear in columns 3 and 4, showed that the differences between sampled and reported hours for each activity were not statistically significant at the 0.05 level for Engineers *A, B, E,* and *H* and for all engineers combined.

It was not possible to check the significance of the differences between sampled and reported activity for Engineer *K* because we only sampled, and he only reported, the single SCR activity, *Project Control.* Qualitatively, however, the large difference between sampled and reported project-control activity suggested a problem with overreporting of administrative activity by this engineer.

Table 2 gives the correlation coefficients between sampled and reported activity hours for Engineers *A, B, E,* and *H* and for all five engineers combined. Except for Engineer *E,* who was tardy in reporting and who filled out his WAR entirely from memory, the generally high coefficients suggested that there was a strong association between sampled and reported activity.

Table 2 — Sampled vs Reported SCR
Activities: Correlation Coefficients*

Timeliness of Reporting	WAR Completed from: Notes (Engineer)	Memory (Engineer)
Prompt	...	0.99 (A) 0.84 (B)
Tardy	0.96 (H)	−0.32 (E)

*All five engineers, 0.73

Table 3 presents the differences between six activity ratios for Engineers *A, B, E* and *H* and for all five engineers combined. The ratios are computed from the sampled and reported data used in Tables 1 and 2. The data for Engineer *K* did not yield meaningful ratios. The results of paired *t* tests showed that the differences between the sampled and reported ratios were not significant at the 0.05 level for Engineers *A, B,* and *H* and for all engineers combined. The differences between the ratios for Engineer *E,* however, were significant at the 0.05 level.

Table 4 gives the correlation coefficients between activity ratios computed from sampled and from reported data for Engineers *A, B, E,* and *H* and for all engineers combined. The coefficients suggested that there was a strong association between sampled and reported ratios both for engineers who were prompt in reporting and for engineers who were tardy but who used notes as memory aids. For the engineer who was tardy and who based his report solely on memory, the coefficient suggested only a chance relationship.

Table 5 presents the differences between sampled and reported data for four activity ratios across engineers. The results of paired *t* tests show that the differences were not significant at the 0.05 level.

Table 3 — Sampled vs Reported SCR Activity Ratio Differences and Paired *t* Test Results by Engineer

SCR Activity Ratio	Engineer				
	A	B	E	H	Combined*
Design/Pseudo Code	0.73	3.33	. . .	. . .	1.3
Design Creating/Design Discussing	8.00	0.5	10.0	−0.37	2.28
Hardware Hiding/Behavior Hiding	. . .	6.08	. . .	. . .	−4.56
Hardware Hiding/Software Decision	−0.10	−1.22	2.0	. . .	0.13
Behavior Hiding/Software Decision	. . .	−1.00	13.0	. . .	0.25
Software Modules/Miscellaneous Activity	. . .	. . .	. . .	9.21	10.06
Mean Difference	2.88	1.54	8.33	4.42	1.58
Computed *t*	1.11	1.10	2.98	0.92	0.81
Critical *t* (0.05)	2.92	2.13	2.92	6.31	2.02

*Engineers A, B, E, H, and K

Table 4 — Sampled vs Reported SCR Activity Ratios: Correlation Coefficients*

Timeliness of Reporting	WAR Completed from:	
	Notes (Engineer)	Memory (Engineer)
Prompt	...	0.99 (A) 0.69 (B)
Tardy	0.99 (H)	0.56 (E)

*All five engineers, 0.38

Table 5 — Sampled vs Reported SCR Activity Ratio Differences and Paired *t* Results Across Engineers

Engineer	Ratio			
	Design/ Pseudo Code	Design Creating/ Design Discussing	Hardware Hiding/ Software Decision	Behavior Hiding/ Software Decision
A	0.73	8.00	−0.10	. . .
B	3.33	0.50	−1.22	−1.00
E	. . .	10.00	2.00	13.00
H	. . .	−0.37	. . .	. . .
K	. . .	. . .	. . .	. . .
Mean Difference	2.03	4.53	0.23	6.00
Computed *t*	1.56	1.72	0.24	0.85
Critical *t* (0.05)	6.34	2.35	2.92	6.34

SUMMARY

The first goal of our study was to determine how accurately engineers record their SCR activities on WARs. Two suppositions were that personnel were grossly inaccurate, and that those who filled out their activity reports promptly were likely to be more accurate than those who were tardy. Paired t test analyses of sampled and reported data showed, however, that although there were some seemingly large differences between sampled and reported data, the differences were not statistically significant. Furthermore, except for one engineer, there was a strong correlation between sampled and reported data. Thus, our conjecture that SCR personnel were grossly inaccurate in reporting activity weekly appears to have been wrong; the engineers were reporting activity with a reasonable degree of accuracy.

The correlation coefficients between sampled and reported activity showed a low correlation coefficient only for the one engineer who completed his activity report late and entirely from memory. This supports our supposition that personnel who filled out their activity reports promptly were likely to be more accurate than those who were tardy. If a person keeps notes of his activity, however, a late-submitted report is not necessarily inaccurate.

Of course, there are two possible objections to these results. The first is that the sampling process, which was run in parallel with the usual reporting of SCR activity on WARS, might have made the weekly reporting more accurate than usual. The second is that the experiment involved only a small experimental group and generated only a small amount of data.

The first objection does not seem to apply to Engineer *H*, who kept an elaborate diary of weekly activity, and to Engineer *K*, who overreported administrative activity, despite having a written record of activity. Conceivably this may apply to Engineers *A*, *B*, and *E*, but we believe that the parallel sampling actually did little to increase the accuracy of their WAR data. Engineer *B* declared just the opposite effect in a follow-up interview. Engineers *A* and *E* did not indicate any effect on their activity reports when questioned.

The second objection is valid. We note, however, that the experimental group consisted, at the time, of almost half the total SCR project personnel. More importantly, the participants were representative of the different ways that engineers completed their WARs.

The second goal of our study was to determine a valid way of analyzing WAR data. Our premise was that the data could be accurately analyzed as ratios of effort. Analyses of differences between selected ratios computed from sampled data and from reported data were conducted for Engineers *A*, *B*, *E*, and *H* and for all five engineers combined. The differences between ratios were not statistically significant for Engineers *A*, *B* and *H* and for all five engineers combined. The differences were significant, however, for engineer *E*, who prepared his activity report solely from memory and submitted it late. These results suggest that activity ratios computed from WAR data are valid. Thus, ratios between activity hours reported on WARs seem to provide useful and practical measures of SCR activity distribution. Computed correlation coeffecients generally support this premise.

The third goal of our study was to evaluate the appropriateness of collecting personnel activity data through weekly activity reports. Our analyses did not indicate significant differences between sampled and reported data or between activity ratios computed from sampled and from reported data. Therefore, it seems that weekly activity reporting is a valid technique for capturing software development activity when reports are promptly submitted. The results presented in Table 2 suggest that prompt weekly reporting, based solely upon memory, is just as accurate as reporting based on notes taken as the work is performed. Problems of inaccuracy are associated only with tardy reporting based solely on memory.

It was surprising to find that 2-hour sampling did not yield significantly different records of project activity compared to the weekly report. The only differences to note are that engineers tended to report slightly more hours on WARs than were captured in sampling, and that the weekly reports showed slightly less detail.

The fourth goal of our study was to determine a useful approach for validating the accuracy of personnel activity data that is collected weekly. The sampling method exhibited several positive benefits and a few drawbacks.

On the positive side, sampling, as done in this study, requires low overhead. It needs only a sampler. Also, surprisingly, the sampled personnel generally do not seem to mind or view it as an interruption to their routine.

On the negative side, some personnel apparently feel awkward when they had no activity to report for a given sampling period. This occurs even though personnel are assured that the sampling does not reflect on their performance ratings in any way. And, as mentioned above, there is also the slight possibility that the sampling process might tend to make the weekly reporting more accurate than usual.

ACKNOWLEDGMENTS

Foremost, we thank the five participants for their cooperation. We also thank our activity samplers, Miss Kathryn Kragh and Ms. Tammy Lewis, who did their job with care and consideration.

There have been many other contributors to our work. The continuing interest of Dr. David Weiss and other NRL research staff members in trying to measure the accuracy of activity data has provided the stimulus for the work reported here. The following persons helped us to improve upon earlier versions of this report: Dr. John O'Hare, Dr. David Weiss, Dr. Henry Ledgard, Mr. Paul Clements, and Ms. Gail Michael.

REFERENCES

V.R. Basili, and D.M. Weiss, "A Methodology for Collecting Valid Software Engineering Data," NRL Report 8679, July 1983.

D.N. Card, F.E. McGarry, G. Page, S. Eslinger, and V.R. Basili, "The Software Engineering Laboratory," NASA Goddard Space Flight Center, Software Engineering Laboratory Series SEL-81-104, February 1982.

W.J. Dixon, and F.J. Massey, Jr., *Introduction to Statistical Analysis* (McGraw-Hill, New York, 1969).

H. Mintzberg, *The Nature of Managerial Work*, (Harper and Row, New York: 1973).

D.L. Parnas, "On the Criteria To Be Used in Decomposing Systems into Modules," *Communications of the ACM*, **15** (12) 1053-1058 (1972).

___, "Use of Abstract Interfaces in the Development of Software for Embedded Computer Systems," NRL Report 8047, June 1977.

APPENDIX A

The Software Cost Reduction (SCR) Project

Since 1978, personnel at the Naval Research Laboratory (NRL) and the Naval Weapons Center (NWC) have been redeveloping version NWC-2 of the operational flight program (OFP) for the A-7E aircraft. They are using such software engineering techniques as information hiding (Parnas 1972), abstract interfaces (Parnas 1977), cooperating sequential processes (Dijkstra 1968), and resource monitors (Hoare 1974). This A-7E OFP redevelopment is currently referred to as the Software Cost Reduction (SCR) project.

The A-7E OFP is part of the Navigation/Weapon Delivery System on the A-7E aircraft. The OFP receives data from sensors, cockpit switches, and a panel from which a pilot keys in data. It controls several displays in the cockpit and positions several sensors. Twenty-two devices are connected to the OFP computer; examples include an inertial measurement set and a head-up display. The inertial measurement set provides velocity data, and the head-up display projects symbols into a pilot's field of view so that he sees them overlaying the view ahead of the aircraft. The OFP calculates navigation information such as present position, speed, and heading; it also controls weapon delivery by giving the pilot steering cues and calculating when to release weapons.

The A-7E OFP is an operational Navy program with severe memory and execution-time constraints. The code consists of approximately 12,000 assembler language instructions for the IBM System 4 PI model TC-2 computer. The TC-2 has 16,000 bytes of memory.

The goals of the SCR project are (1) to demonstrate the feasibility of using the selected software engineering techniques to develop complex, real-time software, and (2) to provide the Navy with a model for the design of avionics software. One of the reasons for choosing to redevelop the A-7E OFP is the challenge of showing that any memory or execution-time overhead incurred by using the software engineering techniques is not prohibitive for such real-time systems. A second reason is that maintenance personnel at NWC feel that the current OFP is difficult to change. The claimed advantage of the selected software engineering techniques is that they facilitate the development of easy-to-change software.

The A-7E software requirements document (Heninger et al. 1978) is the first major product of the SCR project. More recent products of ongoing software design include a guide to OFP software modules (Britton and Parnas 1981), interface specification for the device interface module (Parker et al. 1980), specifications for the function driver module (Clements 1981), specifications for the extended computer module (Britton et al. 1983), and specifications for the shared services module (Clements 1982).

The estimated completion date for the project is September 1985.

REFERENCES

K.H. Britton, and D.L. Parnas, "A-7E Software Module Guide," NRL Memorandum Report 4702, Dec. 1981.

K.H. Britton, D.L. Parnas, and D.M. Weiss, "Interface Specifications for the SCR (A-7E) Extended Computer Module," NRL Memorandum Report 4843, Jan. 1983.

P.C. Clements, "Function Specifications for the A-7E Function Driver Module," NRL Memorandum Report 4658, 1982.

___, "Interface Specifications for the A-7E Shared Services Module," NRL Memorandum Report 4863, Sept. 1982.

E.W. Dijkstra, "Cooperating Sequential Processes," in *Programming Languages,* ed. F. Genuys, (Academic Press, New York, 1968), pp. 43-112.

K.L. Heninger, J.W. Kallander, D.L. Parnas, and J.E. Shore, "Software Requirements for the A-7E Aircraft," NRL Memorandum Report 3876, Nov. 1978.

C.A.R. Hoare, "Monitors: An Operating System Structuring Concept," *Communications of the ACM,* **17** (10) 549-557 (1974).

R.A. Parker, K.L. Heninger, D.L. Parnas, and J.E. Shore, "Abstract Interface Specification for the A-7E Device Interface Module," NRL Memorandum Report 4385, Nov. 1980.

D.L. Parnas, "On the Criteria To Be Used in Decomposing Systems into Modules," *Communications of the ACM,* **15** (12) 1053-1058 (1972).

___, "Use of Abstract Interfaces in the Development of Software for Embedded Computer Systems," NRL Report 8047, June 1977.

APPENDIX B

How SCR Personnel Fill Out Weekly Activity Reports (July 1982)

Engineer A

If the week's activities have been simple, this engineer fills out a weekly activity report (WAR) at the end of the week (i.e., Friday) from memory. Otherwise, he makes notes on a calendar during the week and uses the notes at the end of the week to complete the form. This engineer also uses the form itself to jog his memory regarding activities he might have worked on. Sometimes he checks his completed report to see if the total hours reported can be reconciled with nonproject efforts.

Engineer B

This engineer fills out his report for the week from memory on Friday afternoon (or the following Monday or Tuesday). He uses the form to recall things he might have worked on and reviews other project personnel he might have met with during the week. Before he turns in the completed form, he adds the hours recorded and reconciles the sum with some feeling for what the total sould be.

Engineer C

This engineer responded in writing as follows: "I fill out my WARs at the end of the day or at the end of every other day. I do it from memory by reviewing the day's activities, usually on half-hour units, and assign the period increments to specific activities, e.g., reviewing, discussing details over the phone, testing on the CRT, writing specs, etc. Then at the end of the week (or two), I transcribe these bits and pieces into totals and place them on a form. I do not really check to see whether the figures seem reasonable or not because I have already done that when I 'dice up' my day. The scope of my activities tends to be focused on only one or two general activities, so I do not run into the problem of multitask accounting."

Engineer D

This engineer fills out WARs every week or two. He normally marks his meetings with Engineer E on a calendar and estimates total SCR activity from the recorded meeting times. This engineer has worked most often with Engineer E on the compile-time scheduler, although he is beginning now to review miscellaneous SCR publications.

Engineer E

This engineer is erratic in filling out WARs. For example, he is currently several weeks behind in reporting his work. When he does catch up, he completes the forms from memory. He uses the form itself to trigger his memory, and he considers the various work areas in which he normally is involved. He reviews the completed reports for reasonableness prior to submission.

Engineer F

This engineer records his activity on a calendar throughout the day. On Friday, he translates this record of his work onto a WAR. Before turning in a report, he checks the total hours for reasonableness. This engineer works only two days a week and has only one major activity—testing.

Engineer G

This engineer fills out the WAR from memory at the end of the week. The form serves to jog his memory. He feels that he is accurate because he has the same activities from week to week. He checks each completed form for reasonableness. An interesting note: when this engineer feels that he has not made real progress on some activity, he does not report that effort.

Engineer H

This engineer is erratic in filling out WARs. For example, he currently is several weeks behind in reporting work. However, he usually makes careful notes throughout the day on how he spends his time. He fills out an activity report by transcribing his notes. He sometimes checks the total hours reported on a report for reasonableness.

Engineer I

This engineer records his SCR activity in a notebook at the end of each day or after completing each activity. He often checks his recollection of what he did during the day by discussing classification of activities with Engineer J, with whom he carpools. It is Engineer J who translates this engineer's notebook activities to WARs! This translation occurs every three or four weeks.

Engineer J

This engineer records his activity effort hour-by-hour or activity-by-activity on a WAR throughout the week. When he forgets, he completes his activity report at the end of the day. If later at home he remembers that he forgot to fill out the report during the day, he often makes notes of things for later entry.

Engineer K

This engineer usually fills out a WAR at the end of each week. He notes his non-SCR activities on a calendar or uses a computerized calendar. He uses these notes to fill out a report. He normally has only one SCR activity—project control.

APPENDIX C

Postexperiment Interviews

Engineer A

This engineer felt that sampling every two hours was too distracting. Regarding the accuracy of the weekly activity report (WAR) he submitted for the week, he felt that he 'lost track of detail' because the sampling forced him to dump his memory periodically. He suggested that sampling once a day might be a more realistic interval. He freely admitted that once-a-week reporting is too gross. Indeed, he said that he tended only to report work on software modules at the second level because he is unable to recall relative times spent on lower level modules. But he also expressed hope that an even better method of capturing activities could be found.

Engineer B

This engineer did not mind the interruptions every two hours. He felt, however, that completing the WAR was easier than being sampled every 2 hours. He too, suggested that once-a-day sampling might be a good approach.

Engineer E

This engineer felt that the sampling interval of 2 hours was somewhat irritating. He admitted that the approach was probably more accurate than his usual approach of completing a WAR from memory several weeks later. The sampling experiment made him realize how little work he was devoting to important activities. He now tracks his SCR activity on a form that he updates twice a day.

Engineer H

This engineer did not like the sampling. He often was embarrassed when he had to report no activity to the sampler and had no reason for not having done anything. He felt strongly that the WAR approach is inaccurate, stating that 'no one can remember at the end of the week, unless they write it down.' He concurred with Engineer *A* that he, too, tended to report module activity at the second level when he filled out activitiy reports. He recommended the use of one or two alternative forms for recording activity daily.

Engineer K

This engineer liked the sampling approach and recommended its adoption. He felt it was a more accurate method than the WAR, but cautioned that his work habits were perhaps not typical of the SCR team.

APPENDIX D

Sampled SCR Weekly Activity

SAMPLED

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer A

Date: Friday, 13 August 1982

Activity Area	Activity Hours: Design – Creating	Design – Discussing	Design – Reading	Design – Formal Reviewing	Pseudo Code – Creating	Pseudo Code – Discussing	Pseudo Code – Reading	EC Code – Creating	EC Code – Discussing	EC Code – Reading	Unit Testing
SOFTWARE STRUCTURES											
Module..................											
Process.................											
Uses Hierarchy..........											
SOFTWARE MODULES											
Hardware-Hiding.........											
Extended Computer.......											
Device Interface........							4				
ADC						1	3.5				
Behavior-Hiding.........											
Function Driver.........											
Shared Services.........											
Software Decision.......											
Application Data Type...											
Physical Model..........											
ECM		1.5									
FILT	13										
~~[illegible]~~..........											
WPN	11.5										
System Generation.......											
Software Utility........											
Resource Monitor........											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements...........................	
DIM Specifications.....................	
Function Driver Function Specifications.	
Module Guide...........................	
Project Control........................	
Travel.................................	
Other:	
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

SAMPLED

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer B   Date: Friday, 13 August 1982

Activity Area	Design: Creating	Design: Discussing	Design: Reading	Design: Formal Reviewing	Pseudo Code: Creating	Pseudo Code: Discussing	Pseudo Code: Reading	EC Code: Creating	EC Code: Discussing	EC Code: Reading	Unit Testing
SOFTWARE STRUCTURES											
Module..................											
Process.................											
Uses Hierarchy..........											
SOFTWARE MODULES											
Hardware-Hiding.........											
Extended Computer.......	12										
Device Interface........						1	1				
Behavior-Hiding.........											
Function Driver.........											
Shared Services.........	1.5										
Software Decision.......											
Application Data Type...	1.5										
Physical Model..........	1	2									
Data Banker.............											
System Generation.......											
Software Utility........											
Resource Monitor........											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements..............................	
DIM Specifications........................	
Function Driver Function Specifications.	
Module Guide..............................	
Project Control...........................	0.3
Travel....................................	
Other:	
	0.5
	0.5
organizing	0.5
u Chmura	0.5
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

SAMPLED

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer E

Date: Friday, 13 August 1982

Activity Area	Activity Hours: Design – Creating	Design – Discussing	Design – Reading	Design – Formal Reviewing	Pseudo Code – Creating	Pseudo Code – Discussing	Pseudo Code – Reading	EC Code – Creating	EC Code – Discussing	EC Code – Reading	Unit Testing
SOFTWARE STRUCTURES											
Module											
Process											
Uses Hierarchy											
SOFTWARE MODULES											
Hardware-Hiding											
Extended Computer		1									
Device Interface						1					
Behavior-Hiding											
Function Driver											
Shared Services											
Mode	13										
Software Decision											
Application Data Type											
STE	1										
Physical Model											
Data Banker											
System Generation											
Software Utility											
Resource Monitor											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other (discussing)
program testing				1

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements	
DIM Specifications	
Function Driver Function Specifications	
Module Guide	
Project Control	
Travel	
Other:	
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

SAMPLED

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer H

Date: Friday, 13 August 1982

Activity Area	Activity Hours: Design – Creating	Design – Discussing	Design – Reading	Design – Formal Reviewing	Pseudo Code – Creating	Pseudo Code – Discussing	Pseudo Code – Reading	EC Code – Creating	EC Code – Discussing	EC Code – Reading	Unit Testing
SOFTWARE STRUCTURES											
Module..................											
Process.................											
Uses Hierarchy..........											
SOFTWARE MODULES											
Hardware-Hiding.........											
Extended Computer....											
Device Interface.....											
Behavior-Hiding.........											
Function Driver......											
Shared Services......											
Software Decision......											
Application Data Type											
Physical Model.......											
Data Banker..........											
System Generation....											
AMP.LIB	9.5	4									
Software Utility.....											
Resource Monitor.....											

(See back)

(30 Nov 81, WD-7310a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements..........................	
DIM Specifications....................	
Function Driver Function Specifications.	
Module Guide..........................	
Project Control.......................	
Travel................................	
Other:	
borrowing software for A-7 project	1
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

SAMPLED

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer K   Date: Friday, 13 August 1982

Activity Area	Activity Hours: Design				Pseudo Code			EC Code			Unit Testing
	Creating	Discussing	Reading	Formal Reviewing	Creating	Discussing	Reading	Creating	Discussing	Reading	
SOFTWARE STRUCTURES											
Module...........											
Process...........											
Uses Hierarchy..........											
SOFTWARE MODULES											
Hardware-Hiding.........											
Extended Computer....											
Device Interface......											
Behavior-Hiding.........											
Function Driver......											
Shared Services......											
Software Decision......											
Application Data Type											
Physical Model.......											
Data Banker..........											
System Generation....											
Software Utility.....											
Resource Monitor.....											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements..............................	
DIM Specifications......................	
Function Driver Function Specifications.	
Module Guide..............................	
Project Control..........................	3
Travel....................................	
Other:	
Non A-7 (Optional):	

(30 Nov 81, WD-7310a)

APPENDIX E

Submitted Weekly Activity Reports

rcvd: 17 AUG 1982

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer A **Date:** Friday, 13 August 1982

Activity Area	Activity Hours: Design				Pseudo Code			EC Code			Unit Testing
	Creating	Discussing	Reading	Formal Reviewing	Creating	Discussing	Reading	Creating	Discussing	Reading	
SOFTWARE STRUCTURES											
Module											
Process											
Uses Hierarchy											
SOFTWARE MODULES											
Hardware-Hiding											
Extended Computer											
Device Interface						1	11				
Behavior-Hiding											
Function Driver											
Shared Services											
Software Decision											
Application Data Type											
Physical Model	25	3									
Data Banker											
System Generation											
Software Utility											
Resource Monitor											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements	
DIM Specifications	
Function Driver Function Specifications	
Module Guide	
Project Control	
Travel	
Other:	
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

rcvd: 17 AUG 1982

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer B   Date: Friday, 13 August 1982

Activity Area	Design: Creating	Design: Discussing	Design: Reading	Design: Formal Reviewing	Pseudo Code: Creating	Pseudo Code: Discussing	Pseudo Code: Reading	EC Code: Creating	EC Code: Discussing	EC Code: Reading	Unit Testing
SOFTWARE STRUCTURES											
Module.................											
Process................											
Uses Hierarchy.........											
SOFTWARE MODULES											
Hardware-Hiding........											
Extended Computer......	8	2									
Device Interface.......						1	2				
Behavior-Hiding........											
Function Driver........											
Shared Services........	4										
Software Decision......											
Application Data Type..	3										
Physical Model.........											
Data Banker............											
System Generation......											
Software Utility.......											
Resource Monitor.......											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements...........................	
DIM Specifications.....................	
Function Driver Function Specifications	
Module Guide...........................	
Project Control........................	
Travel.................................	
Other:	
Non A-7 (Optional):	
Charlotte consulting	4

(30 Nov 81, WD-7810a)

rcvd: 3 SEP 1982

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer E

Date: Friday, 13 August 1982

Activity Area	Activity Hours: Design – Creating	Design – Discussing	Design – Reading	Design – Formal Reviewing	Pseudo Code – Creating	Pseudo Code – Discussing	Pseudo Code – Reading	EC Code – Creating	EC Code – Discussing	EC Code – Reading	Unit Testing
SOFTWARE STRUCTURES											
Module........											
Process........											
Uses Hierarchy........											
SOFTWARE MODULES											
Hardware-Hiding........											
Extended Computer........											
Device Interface........											
Behavior-Hiding........											
Function Driver........											
Shared Services........											
Software Decision........											
Application Data Type........											
STE	12	3									
Physical Model........											
Data Banker........											
System Generation........											
Software Utility........											
Resource Monitor........											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements........	
DIM Specifications........	
Function Driver Function Specifications........	
Module Guide........	
Project Control........	
Travel........	
Other:	
process control/management	8
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

rcvd: 10 SEP 1982

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer H

Date: Friday, 13 August 1982

Activity Area	Activity Hours: Design				Pseudo Code			EC Code			Unit Testing
	Creating	Discussing	Reading	Formal Reviewing	Creating	Discussing	Reading	Creating	Discussing	Reading	
SOFTWARE STRUCTURES											
Module..................											
Process.................											
Uses Hierarchy..........											
SOFTWARE MODULES											
Hardware-Hiding.........											
Extended Computer.......											
Device Interface........											
Behavior-Hiding.........											
Function Driver.........											
Shared Services.........											
Software Decision.......											
Application Data Type...											
Physical Model..........											
Data Banker.............											
System Generation.......											
AMP.LIB	11	4									
Software Utility........											
Resource Monitor........											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements..............................	
DIM Specifications........................	
Function Driver Function Specifications.	
Module Guide..............................	
Project Control...........................	
Travel....................................	
Other:	
helping project people with UNIX, C, etc	1
overhead, clean files, etc.	0.5
get multiple bulletin boards working	2
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)

rcvd: 31 AUG 1982

A-7 PROJECT: WEEKLY ACTIVITY REPORT

Your name: Engineer K

Date: Friday, 13 August 1982

Activity Area	Activity Hours: Design – Creating	Design – Discussing	Design – Reading	Design – Formal Reviewing	Pseudo Code – Creating	Pseudo Code – Discussing	Pseudo Code – Reading	EC Code – Creating	EC Code – Discussing	EC Code – Reading	Unit Testing
SOFTWARE STRUCTURES											
Module..................											
Process..................											
Uses Hierarchy..........											
SOFTWARE MODULES											
Hardware-Hiding.........											
Extended Computer.....											
Device Interface......											
Behavior-Hiding.........											
Function Driver.......											
Shared Services.......											
Software Decision.......											
Application Data Type.											
Physical Model........											
Data Banker..........											
System Generation....											
Software Utility.....											
Resource Monitor.....											

(See back)

(30 Nov 81, WD-7810a)

Integration/ Performance Test	Activity Hours: Designing	Performing	Reviewing	Other

Miscellaneous Activity	Hours
Documentation Maintenance	
Requirements............................	
DIM Specifications......................	
Function Driver Function Specifications.	
Module Guide............................	
Project Control..........................	16
Travel...................................	
Other:	
Non A-7 (Optional):	

(30 Nov 81, WD-7810a)